Claims:

1.   A method of parsing a markup language document comprising syntactic elements, said method comprising, for one of said syntactic elements, the steps of:

5   identifying a type of the element;

processing the element by determining a hash representation thereof if said type is a first type; and

augmenting an at least partial structural representation of the document using the hash representation if said type is said first type.

10

2.   A method according to claim 1, wherein said parsing is event-based parsing.

3.   A method according to claim 1, wherein said hash representation is

15   determined using one of:

a hash algorithm;

a first reference to said hash algorithm dependent upon an associated Universal Reference Indicator;

a second reference to said hash algorithm dependent upon an associated

20   namespace; and

a third reference to said hash algorithm dependent upon an associated Extended Markup Language declaration;

4.   A method according to claim 2, wherein said first type is one of:

25   one of a structural element and a part thereof;

a definition of said structural element;

a declaration of said structural element; and

a match for said structural element.

5.   A method according to claim 4, wherein said structural element is a tag.

6.   A method according to claim 2, wherein the hash representation is a unique code for said one syntactic element, said element having less than a first number of characters.

7.   A method according to claim 2, wherein the hash representation is not a unique code for said one syntactic element, said element being constrained, to a probability level, in terms of at least one of (i) a number of characters in the element and (ii) a permissible number of permutations of characters in the element.

8.   A method according to claim 6, wherein said code comprises numeric characters.

9.   A method according to claim 2, wherein said processing step comprises a sub-step of:

determining an extended hash representation of both (i) said one syntactic element being a first instance of said first type, and (ii) another syntactic element being a second instance of said first type, within which said first instance, said second instance is nested.

10.   A method according to claim 1 comprising, for another one of said syntactic elements, further steps of:

identifying a type of the other element; and if the type of the other element is equivalent to said first type:

(i) processing the other element to thereby determine a second hash representation thereof; and

(ii) augmenting said at least partial structural representation of the document using the second hash representation, wherein:

said processing and said second processing ensure that if a first relationship exists between the one element and the other element, then a second relationship which is representative of the first relationship, exists between the hash representation of the one element and the hash representation of the other element.

11.   A method according to claim 10, wherein:

the one element is a start tag;

the other element is an end tag;

the hash representation of the one element is a corresponding hashed start tag, and;

the second hash representation of the other element is a corresponding hashed end tag.

12.   A method according to claim 11, wherein:

the end tag is a first modification of the start tag; and

the hashed end tag is a second modification of the hashed start tag, said second modification being representative of the first modification.

13.   A method according to claim 12, wherein:

the end tag is the same as the start tag apart from having a distinguishing character incorporated therein; and

the hashed end tag is at least one of:

the same as the hashed start tag;

the same as the hashed start tag apart from having a distinguishing character incorporated therein; and

the hashed start tag having been processed by an operator.

14.   A method according to claim 12, wherein:

the one and the other element comprise respectively a start tag and an end tag, being a first pair of tags;

corresponding hashed start and end tags for said first pair of tags are incorporated into the partial structural representation of said document;

5   the document further includes a second pair of tags comprising a respective start tag and end tag, said second pair of tags being nested within said first pair of tags in said document, said method comprising further steps of:

processing said second pair of tags to form corresponding second hashed start and end tags;

10   augmenting said at least partial structural representation of the document using said corresponding second hashed start and end tags so that said second hashed start and end tags indicate a nesting in relation to said hashed start and end tags for the first pair of tags which is equivalent to the nesting of said second pair of tags within said first pair of tags.

15

15.   A method according to claim 14 comprising, prior to said augmenting step, a further step of:

concatenating the first hashed start tag with the second hashed start tag, and concatenating the first hashed end tag with the second hashed end tag, to thereby form

20   respective extended hashed start and end tags for said second pair, wherein:

said augmenting step is performed using said respective extended hashed start and end tags for said second pair, and:

said extended hashed start and end tags indicate a nesting in relation to said hashed start and end tags for the first pair of tags which is equivalent to the nesting of said

25   second pair of tags within said first pair of tags.

16.   A method according to claim 1, wherein the augmenting step is succeeded by a well-formedness checking step against a syntactic rule, said well-formedness checking step comprising checking said at least partial structural

representation of the markup language document against the syntactic rule by numerically comparing corresponding hashed representations of elements in said at least partial structural representation of the markup language document

17. A method according to claim 16, wherein said numerically comparing step is succeeded by a further step of:

string-comparing, in accordance with said syntactic rule, corresponding non-hashed representations of elements not of said first type.

18. A method according to claim 1, wherein said first type is one of:

one of a structural element and a part thereof;

a definition of said structural element;

a declaration of said structural element; and

a match of said structural element.

19. A method according to claim 18, wherein said structural element is a tag.

20. A method according to claim 14, comprising a further step of:

checking the well-formedness of said at least partial structural representation of the document against a syntactic rule.

21. A method according to claim 20, wherein the syntactic rule relates to proper nesting of tags and said checking step comprises sub-steps of:

performing a numerical comparison across hashed tags in said at least partial structural representation of the document to thereby identify said first hashed start and end tags and said second hashed start and end tags; and

verifying that the second hashed start and end tags indicate a proper nesting in relation to said first hashed start and end tags.

22.   A method according to claim 21, wherein the numerical comparison is followed by a further step of:

performing a string comparison, in accordance with said syntactic rule, across non-hashed parts of respective tags in said at least partial structural representation of the document.

23.   A method according to claim 15, comprising a further step of:

checking the well-formedness of said at least partial structural representation of the document against a syntactic rule.

24.   A method according to claim 23, wherein the syntactic rule relates to proper nesting of tags and said checking step comprises sub-steps of:

performing a numerical comparison across hashed tags in said at least partial structural representation of the document to thereby identify said first hashed start and end tags and said extended hashed start and end tags; and

verifying that the extended hashed start and end tags indicate a proper nesting in relation to said first hashed start and end tags.

25.   A method according to claim 24, wherein the numerical comparison is followed by a further step of:

performing a string comparison across non-hashed parts of respective tags in said at least partial structural representation of the document.

26.   A method according to claim 16, wherein the well-formedness checking step is one of (a) succeeded by, (b) included in, and (c) replaced by a validation step against a validation reference document VRD, said validation step comprising sub-steps of:

(a) processing the VRD, said processing comprising, for a syntactic element in the VRD, sub-sub-steps of:

(i) identifying a type of said syntactic element of the VRD; and

(ii) processing the syntactic element by determining a hash representation thereof if said type is said first type; and

(b) checking said at least partial structural representation of the markup language document against the processed VRD, said checking comprising a sub-sub-step of numerically comparing corresponding hashed representations of elements.

27. A method of validating a markup language document against a VRD, said method comprising steps of:

(a) processing the markup language document, for each document tag identified therein, if said document tag is not a first document tag in a corresponding markup language document tag hierarchy, said processing comprising steps of:

(i) determining a hierarchy position of said document tag;

(ii) determining an extended hashed representation of said document tag concatenated with a hashed representation of a previous document tag in the document tag hierarchy; and

(iii) storing said extended hashed representation of said document tag if said document tag is more deeply nested than a previous document tag;

(b) processing said VRD, for each tag identified therein, if said tag is not a first tag in a corresponding tag hierarchy, said processing comprising steps of:

(i) determining a hierarchy position of said tag;

(ii) determining an extended hashed representation of said tag concatenated with a hashed representation of a previous tag in the corresponding tag hierarchy; and

(iii) storing said extended hashed representation of said tag in a list; and

(c) validating said markup language document if said extended hashed representation of said document tag is one of found in said list and is a valid subset of a member of said list.

28. A method of validating a markup language document against a VRD, said method comprising steps of:

(a) processing said VRD, for each structural element identified therein, said processing comprising steps of:

(i) determining syntactic attributes of said structural element;

(ii) determining a hashed representation of said structural element; and

(iii) storing said hashed representation and syntactic attributes of said structural element in a structural representation of said VRD; and

(b) processing the markup language document, for each document structural element identified therein, said processing comprising steps of:

(i) determining syntactic attributes of said document structural element;

(ii) determining a hashed representation of said document structural element; and

(iii) storing said hashed representation and syntactic attributes of said document structural element in a structural representation of the document; and

(c) validating said markup language document if syntactic attributes and hashed representations of said each document structural element in the structural representation of the document conforms to corresponding syntactic attributes and hashed representations in said structural representation of said VRD.

29. A method according to claim 26, wherein said numerically comparing step is succeeded by a further step of string-comparing corresponding non-hashed representations of elements not of said first type.

30. A method according to claim 26, wherein said first type is one of:

one of a structural element and a part thereof;

a definition of said structural element;

a declaration of said structural element; and

a match of said structural element.

31.   A method according to claim 30, wherein said structural element is a tag.

32.   A method of encoding a markup language document comprising syntactic elements, said method comprising, for one of said syntactic elements, steps of:

identifying a type of the syntactic element; and

processing the syntactic element by one of:

(i)   determining a hash representation thereof if said type is a first type;

(ii)   determining a compressed representation thereof if said type is not a first type; and

(iii) retaining the syntactic element.

33.   A method of decoding a markup language document comprising encoded syntactic elements, said method comprising, for one of said encoded syntactic elements, steps of:

identifying a type of the encoded syntactic element;

processing the encoded syntactic element by at least one of:

(i)   determining an inverse hash representation thereof if said type is a first type; and

(ii)   determining a decompressed representation thereof if said type is not a first type; and

(iii) retaining the encoded syntactic element.

34.   An apparatus for parsing a markup language document comprising syntactic elements, said apparatus comprising:

identifying means for identifying a type of the element;

processing means for processing the element by determining a hash

5   representation thereof if said type is a first type; and

augmenting means for augmenting an at least partial structural representation of the document using the hash representation if said type is said first type.

35.   An apparatus for validating a markup language document against a

10   VRD, said apparatus comprising:

(a) means for processing the markup language document, for each document tag identified therein, if said document tag is not a first document tag in a corresponding markup language document tag hierarchy, said means comprising:

(i) means for determining a hierarchy position of said document tag;

15   (ii) means for determining an extended hashed representation of said document tag concatenated with a hashed representation of a previous document tag in the document tag hierarchy; and

(iii) means for storing said extended hashed representation of said document tag if said document tag is more deeply nested than an extended hashed

20   representation of a previous document tag;

(b) means for processing said VRD, for each tag identified therein, if said tag is not a first tag in a corresponding tag hierarchy, said means comprising:

(i) means for determining a hierarchy position of said tag;

(ii) means for determining an extended hashed representation of said tag

25   concatenated with a hashed representation of a previous tag in the corresponding tag hierarchy; and

(iii) means for storing said extended hashed representation of said tag in a list; and

(c) means for establishing whether said extended hashed representation of said document tag is one of to be found in said list, and is a valid subset of a member of said list, thereby validating said markup language document.

36. An apparatus for validating a markup language document against a VRD, said apparatus comprising:

(a) means for processing said VRD, for each structural element identified therein, said means comprising:

(i) means for determining syntactic attributes of said structural element;

(ii) means for determining a hashed representation of said structural element; and

(iii) means for storing said hashed representation and syntactic attributes of said structural element in a structural representation of said VRD; and

(b) means for processing the markup language document, for each document structural element identified therein, said means comprising:

(i) means for determining syntactic attributes of said document structural element;

(ii) means for determining a hashed representation of said document structural element; and

(iii) means for storing said hashed representation and syntactic attributes of said document structural element in a structural representation of the document; and

(c) means for comparing syntactic attributes and hashed representations of said each document structural element in the structural representation of the document to corresponding syntactic attributes and hashed representations in said structural representation of said VRD to thereby establish validity of the markup language document.

37.   An apparatus for encoding a markup language document comprising syntactic elements, to form an at least partial structural representation of the document, said apparatus comprising:

means for identifying a type of the syntactic element; and

means for processing the syntactic element by one of:

(i)   determining a hash representation thereof if said type is a first type;

(ii)   determining a compressed representation thereof if said type is not a first type; and

(iii) retaining the syntactic element.

38.   An apparatus for decoding a markup language document comprising encoded syntactic elements, said apparatus comprising:

means for identifying a type of the encoded syntactic element;

means for processing the encoded syntactic element by at least one of:

(i)   determining an inverse hash representation thereof if said type is a first type;

(ii)   determining a decompressed representation thereof if said type is not a first type; and

(iii) retaining the encoded syntactic element.

39.   A computer program which is configured to make a computer execute a procedure to parse a markup language document comprising syntactic elements, said program comprising:

code for identifying a type of an element;

code for processing the element by determining a hash representation thereof if said type is a first type; and

code for augmenting an at least partial structural representation of the document using the hash representation if said type is said first type.

40.   A computer program which is configured to make a computer execute a procedure to validate a markup language document against a VRD, said program comprising:

(a) code for processing the markup language document, for each document tag identified therein, if said document tag is not a first document tag in a corresponding markup language document tag hierarchy, said code comprising:

(i) code for determining a hierarchy position of said document tag;

(ii) code for determining an extended hashed representation of said document tag concatenated with a hashed representation of a previous document tag in the document tag hierarchy; and

(iii) code for storing said extended hashed representation of said document tag if said tag is more deeply nested than a previous document tag;

(b) code for processing said VRD, for each tag identified therein, if said tag is not a first tag in a corresponding tag hierarchy, said code comprising:

(i) code for determining a hierarchy position of said tag;

(ii) code for determining an extended hashed representation of said tag concatenated with a hashed representation of a previous tag in the corresponding tag hierarchy; and

(iii) code for storing said extended hashed representation of said tag in a list; and

(c) code for validating said markup language document if said extended hashed representation of said document tag is one of found in said list, and is a valid subset of a member of said list.

41.   A computer program which is configured to make a computer execute a procedure to validate a markup language document against a VRD, said program comprising:

(a) code for processing said VRD, for each structural element identified therein, said code comprising:

(i) code for determining syntactic attributes of said structural element;

(ii) code for determining a hashed representation of said structural element; and

(iii) code for storing said hashed representation and syntactic attributes of said structural element in a structural representation of said VRD; and

(b) code for processing the markup language document, for each document structural element identified therein, said code comprising:

(i) code for determining syntactic attributes of said document structural element;

(ii) code for determining a hashed representation of said document structural element; and

(iii) code for storing said hashed representation and syntactic attributes of said document structural element in a structural representation of the document; and

(c) code for validating said markup language document if syntactic attributes and hashed representations of said each document structural element in the structural representation of the document conforms to corresponding syntactic attributes and hashed representations in said structural representation of said VRD.

42.   A computer program which is configured to make a computer execute a procedure to encode a markup language document comprising syntactic elements, said program comprising:

code for identifying a type of the syntactic element; and

code for processing the syntactic element by one of:

(i)   determining a hash representation thereof if said type is a first type;

(ii) determining a compressed representation thereof if said type is not a first type; and

(iii) retaining the syntactic element.

43.   A computer program which is configured to make a computer execute a procedure to decode a markup language document comprising encoded syntactic elements, said program comprising:

code for identifying a type of the encoded syntactic element;

code for processing the encoded syntactic element by at least one of:

(i)   determining an inverse hash representation thereof if said type is a first type; and

(ii) determining a decompressed representation thereof if said type is not a first type; and

(iii) retaining the encoded syntactic element.

44.   A computer program product including a computer readable medium having recorded thereon a computer program which is configured to make a computer execute a procedure to parse a markup language document, said program comprising:

code for identifying a type of the element;

code for processing the element by determining a hash representation thereof if said type is a first type; and

code for augmenting an at least partial structural representation of the document using the hash representation if said type is said first type.

45.   A computer program product including a computer readable medium having recorded thereon a computer program which is configured to make a computer execute a procedure to validate a markup language document against a VRD, said program comprising:

(a) code for processing the markup language document, for each document tag identified therein, if said document tag is not a first document tag in a corresponding markup language document tag hierarchy, said code comprising:

(i) code for determining a hierarchy position of said document tag;

(ii) code for determining an extended hashed representation of said document tag concatenated with a hashed representation of a previous document tag in the document tag hierarchy; and

(iii) code for storing said extended hashed representation of said document tag if said document tag is more deeply nested than a previous document tag;

(b) code for processing said VRD, for each tag identified therein, if said tag is not a first tag in a corresponding tag hierarchy, said code comprising:

(i) code for determining a hierarchy position of said tag;

(ii) code for determining an extended hashed representation of said tag concatenated with a hashed representation of a previous tag in the corresponding tag hierarchy; and

(iii) code for storing said extended hashed representation of said tag in a list; and

(c) code for validating said markup language document if said extended hashed representation of said document tag is one of found in said list and is a valid subset of a member of said list.

46.   A computer program product including a computer readable medium having recorded thereon a computer program which is configured to make a computer execute a procedure to validate a markup language document against a VRD, said program comprising:

(a) code for processing said VRD, for each structural element identified therein, said code comprising:

(i) code for determining syntactic attributes of said structural element;

(ii) code for determining a hashed representation of said structural element; and

(iii) code for storing said hashed representation and syntactic attributes of said structural element in a structural representation of said VRD; and

(b) code for processing the markup language document, for each document structural element identified therein, said code comprising:

(i) code for determining syntactic attributes of said document structural element;

(ii) code for determining a hashed representation of said document structural element; and

(iii) code for storing said hashed representation and syntactic attributes of said document structural element in a structural representation of the document; and

(c) code for validating said markup language document if syntactic attributes and hashed representations of said each document structural element in the structural representation of the document conforms to corresponding syntactic attributes and hashed representations in said structural representation of said VRD.

47. An at least partial structural representation a markup language document comprising syntactic elements, said at least partial representation having been produced by a method comprising, for one of said syntactic elements, the steps of:

identifying a type of the element;

processing the element by determining a hash representation thereof if said type is a first type; and

augmenting an at least partial structural representation of the document using the hash representation if said type is said first type.

48. An apparatus for parsing a markup language document comprising syntactic elements, said apparatus comprising:

a processor;

a memory for storing (i) the document, and (ii) a program which is configured to make the processor execute a procedure to parse the document;

said program comprising:

(i) code for identifying a type of an element;

(ii) code for processing the element by determining a hash representation thereof if said type is a first type; and

(iii) code for augmenting an at least partial structural representation of the document using the hash representation if said type is said first type.

49.   An apparatus for validating a markup language document comprising syntactic elements against a VRD comprising syntactic elements, said apparatus comprising:

(a) a processor;

(b) a memory for storing (i) the document, (ii) said VRD, and (iii) a program which is configured to make the processor execute a procedure to validate the document;

(c) said program comprising:

(ca) code for processing the markup language document, for each document tag identified therein, if said document tag is not a first document tag in a corresponding markup language document tag hierarchy, said code comprising:

(caa) code for determining a hierarchy position of said document tag;

(cab) code for determining an extended hashed representation of said document tag concatenated with a hashed representation of a previous document tag in the document tag hierarchy; and

(cac) code for storing said extended hashed representation of said document tag if said document tag is more deeply nested than a previous document tag;

(cb) code for processing said VRD, for each tag identified therein, if said tag is not a first tag in a corresponding tag hierarchy, said means comprising:

(cba) code for determining a hierarchy position of said tag;

(cbb) code for determining an extended hashed representation of said tag concatenated with a hashed representation of a previous tag in the corresponding tag hierarchy; and

(cbc) code for storing said extended hashed representation of said tag in a list; and

(cc) code for establishing whether said extended hashed representation of said document tag is one of to be found in said list, and is a valid subset of a member

5   of said list, thereby validating said markup language document.

50.   An apparatus for validating a markup language document containing syntactic elements against a VRD containing syntactic elements, said apparatus comprising:

10   (a) a processor;

(b) a memory for storing (i) the document, (ii) said VRD, and (iii) a program which is configured to make the processor execute a procedure to validate the document;

(c) said program comprising:

(ca) code for processing said VRD, for each structural element

15   identified therein, said code comprising:

(caa) code for determining syntactic attributes of said structural element;

(cab) code for determining a hashed representation of said structural element; and

20   (cac) code for storing said hashed representation and syntactic attributes of said structural element in a structural representation of said VRD; and

(cb) code for processing the markup language document, for each document structural element identified therein, said code comprising:

(caa) code for determining syntactic attributes of said

25   document structural element;

(cab) code for determining a hashed representation of said document structural element; and

(cac) code for storing said hashed representation and syntactic attributes of said document structural element in a structural representation of the document; and

(cc) code for comparing syntactic attributes and hashed representations of said each document structural element in the structural representation of the document to corresponding syntactic attributes and hashed representations in said structural representation of said VRD to thereby establish validity of the markup language document.

51.   A method of validating a markup language document against a VRD, said method comprising steps of:

determining first extended hashed representation(s) for most deeply nested syntactic element(s) of a first type in the VRD;

storing said first extended hashed representation(s) in a VRD list;

determining a second extended hashed representation for a most deeply nested syntactic element of the first type in the markup language document; and

declaring said markup language document to not be invalid if said second extended hashed representation is present in the VRD list.

52.   A method according to claim 51, wherein said syntactic element of said first type is one of:

one of a structural element and a part thereof;

a definition of said structural element; ·

a declaration of said structural element; and

a match for said structural element.

53.   A method according to claim 52, wherein said structural element is a tag.

54. A method according to claim 51, wherein:

said most deeply nested syntactic elements(s) in the VRD are syntactic elements(s) which are most deeply nested within one of the global structure of the VRD and a local sub-structure of the VRD; and

5   said most deeply nested syntactic element in the markup language document is a syntactic element which is most deeply nested within one of the global structure of the markup language document and a local sub-structure of the markup language document.

55. An apparatus for validating a markup language document against a
10   VRD, said apparatus comprising:

means for determining first extended hashed representation(s) for most deeply nested syntactic element(s) of a first type in the VRD;

means for storing said first extended hashed representation(s) in a VRD list;

means for determining a second extended hashed representation for a most
15   deeply nested syntactic element of the first type in the markup language document; and

means for declaring said markup language document to not be invalid if said second extended hashed representation is present in the VRD list.

56. A computer program which is configured to make a computer execute a
20   procedure to validate a markup language document against a VRD, said program comprising:

code for determining first extended hashed representation(s) for most deeply nested syntactic element(s) of a first type in the VRD;

code for storing said first extended hashed representation(s) in a VRD list;

25   code for determining a second extended hashed representation for a most deeply nested syntactic element of the first type in the markup language document; and

code for declaring said markup language document to not be invalid if said second extended hashed representation is present in the VRD list.

57. A computer program product including a computer readable medium having recorded thereon a computer program which is configured to make a computer execute a procedure to validate a markup language document against a VRD, said program comprising:

5   code for determining first extended hashed representation(s) for most deeply nested syntactic element(s) of a first type in the VRD;

code for storing said first extended hashed representation(s) in a VRD list;

code for determining a second extended hashed representation for a most deeply nested syntactic element of the first type in the markup language document; and

10   code for declaring said markup language document to not be invalid if said second extended hashed representation is present in the VRD list.

58. A method according to claim 7, wherein said code comprises numeric characters.

15